EXHIBIT B

```
0 #include <stdio.h>
1
2 /* #define HYPER_DEBUG 1 */
3
4 /*  ********************************************************************
5     File: sslctx.c
6
7     SSL Plus: Security Integration Suite(tm)
8     Version 1.1.1 -- August 11, 1997
9
10    Copyright (c)1996, 1997 by Consensus Development Corporation
11         Copyright (c)1997, 1998 by Aventail Corporation
12
13    Portions of this software are based on SSLRef(tm) 3.0, which is
14    Copyright (c)1996 by Netscape Communications Corporation. SSLRef(tm)
15    was developed by Netscape Communications Corporation and Consensus
16    Development Corporation.
17
18    In order to obtain this software, your company must have signed
19    either a PRODUCT EVALUATION LICENSE (a copy of which is included in
20    the file "LICENSE.TXT"), or a PRODUCT DEVELOPMENT LICENSE. These
21    licenses have different limitations regarding how you are allowed to
22    use the software. Before retrieving (or using) this software, you
23    *must* ascertain which of these licenses your company currently
24    holds. Then, by retrieving (or using) this software you agree to
25    abide by the particular terms of that license. If you do not agree
26    to abide by the particular terms of that license, than you must
27    immediately delete this software. If your company does not have a
28    signed license of either kind, then you must either contact
29    Consensus Development and execute a valid license before retrieving
30    (or using) this software, or immediately delete this software.
31
32    ********************************************************************
33
34    File: sslctx.c      SSLContext accessors
35
36    Functions called by the end user which configure an SSLContext
37    structure or access data stored there.
38
```

<because of the size of this file and the fact that most of it is not relevant, we include only the relevant function SSLSetSloppyMode here>

```
899 SSLErr CDECL SSLSetSloppyMode(SSLContext *ctx, int m)
900 {
901    if(ctx == NULL)
902         return SSLUnknownErr;
903
904    if((ctx->selectedCipherSpec->cipher->blockSize == 0) &&
905       (ctx->selectedCipherSpec->cipher->keySize > 0))
906         return SSLProtocolErr;
907
908    ctx->ssloppy = m;
909
910    return SSLNoErr;
911 }
```